The Tradeoff Between Cyclic Topology and Complexity in Graphical Models of Linear Codes

Thomas R. Halford and Keith M. Chugg
Communication Sciences Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2565
Email: {halford,chugg}@usc.edu

Abstract—It is now well-known that together with a suitable message passing schedule, a graphical model for a code implies a soft-in soft-out (SISO) decoding algorithm which is optimal for cycle-free models and suboptimal, yet often substantially less complex, for cyclic models. The Cut-Set Bound (CSB) indicates that the introduction of cycles to graphical models can lead to dramatic reductions in complexity and in the case of tail-biting trellises (i.e. single-cycle models), the CSB can be used to establish the square-root bound, thus quantifying this potential reduction precisely. The aim of this work is to provide such a precise characterization of the tradeoff between cyclic topology and complexity (i.e. maximum hidden variable size and local constraint complexity) in graphical models with more than one cycle. To this end, a new bound is introduced - the tree-inducing cut-set bound - which can be viewed as a generalization of the square-root bound to graphical models with arbitrary cyclic topologies.

I. INTRODUCTION

Graphical models of codes have been studied since the 1960s and this study has intensified in recent years due to the discovery of turbo codes by Berrou et al. [1], the rediscovery of Gallager’s low-density parity-check (LDPC) codes [2] by Spielman et al. [3] and MacKay et al. [4], and the pioneering work of Wiberg, Loeliger and Koetter [5], [6]. It is now well-known that together with a suitable message passing schedule, a graphical model implies a SISO decoding algorithm which is optimal for cycle-free models and suboptimal, yet often substantially less complex, for cyclic models (cf. [6], [7], [8], [9], [10]). It has been observed empirically in the literature that there exists a correlation between the cyclic topology of a graphical model and the performance of the decoding algorithms implied by that graphical model (cf. [5], [10], [11], [12], [13], [14], [15], [16]). To summarize this empirical “folk-knowledge”, those graphical models which imply near-optimal decoding algorithms tend to have large girth, a small number of short cycles and a cycle structure implying near-optimal decoding algorithms tend to have large girth, a small number of short cycles and a cycle structure that is not overly regular.

Two broad classes of graphical modeling problems can be identified in the literature:

• Constructive problems: Given a set of design requirements, design a suitable code by constructing a good graphical model (i.e. a model which implies a low-complexity, near-optimal decoding algorithm).
• Extractive problems: Given a specific code, extract a graphical model for that code which implies a decoding algorithm with desired complexity and performance characteristics.

Constructive graphical modeling problems have been widely addressed by the coding theory community. Capacity approaching LDPC codes have been designed for both the additive white Gaussian noise channel (cf. [17], [18]) and the binary erasure channel (cf. [19], [20]). Other classes of modern codes have been successfully designed for a wide range of practically motivated block lengths and rates (cf. [21], [22], [23], [24], [25]).

Less is understood about extractive graphical modeling problems, however. The extractive problems that have received the most attention are those concerning Tanner graph [11] and trellis representations of block codes. Tanner graphs imply low-complexity decoding algorithms; however, the Tanner graphs corresponding to many block codes of practical interest, e.g. high-rate Reed-Muller (RM), Reed-Solomon (RS), and Bose-Chaudhuri-Hocquenghem (BCH) codes, necessarily contain many short cycles [26] and thus imply poorly performing decoding algorithms. There is a well-developed theory of conventional trellises [27] and tail-biting trellises [28], [29] for linear block codes. Conventional and tail-biting trellises imply optimal and, respectively, near-optimal decoding algorithms; however, for many block codes of practical interest these decoding algorithms are prohibitively complex thus motivating the study of more general graphical models (i.e. models with a richer cyclic topology than a single cycle).

The tradeoff between cyclic topology and complexity can be observed in graphical models for both modern and classical codes. Averaged over all interleavers, the full trellis of Berrou et al.’s original turbo code [1] with interleaver size $K$ contains hidden (state) variables with alphabet size $2^{K/2}$ [30]. These codes, however, have well-known graphical models composed of two parallel trellises containing only 16-ary hidden variables connected by an interleaver which introduces $\geq K^2/2$ cycles. The minimal bit-level trellis of the binary extended Golay code contains a 256-ary hidden variable at its midpoint [31]. By introducing a single cycle, a tail-biting trellis containing only 16-ary hidden variables can be obtained for this code [28].

Wiberg et al. first described the tradeoff between cyclic topology and complexity in graphical models for codes...
and established the Cut-Set Bound in [5], [6]. In his first Codes on Graphs paper [10], Forney provided the following description of this tradeoff (Sec. IV.G):

On cycle-free graphs, each edge is a cut set, and the size of the state space associated with that edge is lower-bounded by the Cut-Set Bound to be at least as great as that of a state space in some conventional state realization (trellis). Therefore, no dramatic reduction in complexity over conventional realizations can be expected by using more general cycle-free realizations.

On the other hand, on graphs with cycles, cut sets generally comprise multiple state variables, and the minimum complexity implied by the Cut-Set Bound may be spread across these variables; therefore, dramatic reductions in state complexity may be obtained.

The Cut-Set Bound provides a precise characterization of the potential tradeoff between graphical model complexity and cyclic topology in the case of tail-biting trellises (i.e. single-cycle models). Specifically, the CSB can be used to establish the square-root bound [28]: the maximum hidden variable alphabet size in a tail-biting trellis for a given code $C$ is lower-bounded by the square-root of the minimum possible midpoint hidden variable alphabet size of a conventional trellis for $C$. The aforementioned tail-biting trellis for the Golay code meets this bound [28]. As was first noted by Wiberg et al. [5], it is very challenging, however, to use the CSB to characterize the tradeoff between cyclic topology and complexity for graphical models with cyclic topologies richer than a single cycle.

The primary goal of this work is to provide a more precise characterization of the potential tradeoff between graphical model complexity and cyclic topology than that offered by the Cut-Set Bound alone. To this end this work introduces a new bound - the tree-inducing cut-set bound - which may be viewed as a generalization of the square-root bound for tail-biting trellises to graphical models with arbitrary cyclic topologies. Inasmuch as the cyclic topology of a graphical model is related to the performance of the decoding algorithms it implies, the bound presented in this work provides insight into the limits of graphical model extraction.

The remainder of this work is organized as follows. Section II introduces notation used throughout while Section III reviews graphical models for codes. The main result is stated and proved in Section IV and discussed in Section V. Concluding remarks are given in Section VI.

II. NOTATION AND PRELIMINARIES

A. Notation

The set of consecutive integers is denoted

$$[a, b] = \{a, a + 1, \ldots, b - 1, b\} \subset \mathbb{Z}. \quad (1)$$

The binomial coefficient is denoted $\binom{a}{b}$. The finite field with $q$ elements is denoted $\mathbb{F}_q$ and has addition operator $+$. Given a finite index set $I$, the vector space over $\mathbb{F}_q$ defined on $I$ is

$$\mathbb{F}_q^I = \{f = (f_i \in \mathbb{F}_q, i \in I)\}. \quad (2)$$

Suppose that $J \subseteq I$ is some subset of the index set $I$. The projection of a vector $f \in \mathbb{F}_q^I$ onto $J$ is denoted

$$f_{|J} = (f_i, i \in J). \quad (3)$$

B. Codes, Projections and Subcodes

Given a finite index set $I$, a linear code over $\mathbb{F}_q$ defined on $I$ is some vector subspace $C \subseteq \mathbb{F}_q^I$. The block length and dimension of $C$ are denoted $n(C) = |J|$ and $k(C) = \dim C$, respectively. Often $I$ is chosen as $[1, n(C)]$; however, such an ordering on $I$ need not be assumed in this work. A code $C$ can be described by a $k(C) \times n(C)$ generator matrix over $\mathbb{F}_q$, $G_C$, whose rows span $C$. The null space of a code $C$ is its dual code $C^\perp$ with block length $n(C^\perp) = n(C)$ and dimension $k(C^\perp) = n(C) - k(C)$. A generator matrix for $C^\perp$ is also a parity-check matrix for $C$, $H_C$.

Given a subset $J \subseteq I$, the projection of $C$ onto $J$ is the set of all codeword projections: $C_{|J} = \{c_{|J}, c \in C\}$. Closely related to $C_{|J}$ is the subcode $C_J$: the projection onto $J$ of the subset of codewords satisfying $c_i = 0$ for $i \in I \setminus J$. Both $C_{|J}$ and $C_J$ are linear codes.

Finally, suppose that $C_1$ and $C_2$ are two codes over $\mathbb{F}_q$ defined on the same index set $I$. The intersection of $C_1$ and $C_2$, $C_1 \cap C_2$, is a linear code defined on $I$ comprising the vectors in $\mathbb{F}_q^I$ that are contained in both $C_1$ and $C_2$.

C. Graph Theory

A graph $G = (V, E, \mathcal{H})$ consists of:

- A finite non-empty set of vertices $V$.
- A set of edges $E$, which is some subset of the pairs $\{u, v\} : u, v \in V, u \neq v$.
- A set of half-edges $\mathcal{H}$, which is any subset of $V$.

It is non-standard to define graphs with half-edges; however, as will be demonstrated in Section III, half-edges are useful in the context of graphical models for codes. A walk of length $n$ in $G$ is a sequence of vertices $v_1, v_2, \ldots, v_n, v_{n+1}$ in $V$ such that $\{v_i, v_{i+1}\} \in E$ for all $i \in [1, n]$. A path is a walk on distinct vertices while a cycle of length $n$ is a walk such that $v_1$ through $v_n$ are distinct and $v_1 = v_{n+1}$. A tree is a graph containing no cycles (i.e. a cycle-free graph). A graph is connected if any two of its vertices are linked by a walk. A cut in a connected graph $G$ is some subset of edges, $\mathcal{X} \subseteq E$, whose removal yields a disconnected graph. Cuts thus partition the vertex set $V$. This work is especially concerned with a specific type of cut:

**Definition 1:** Let $G$ be a connected graph. A tree-inducing cut is some subset of edges, $\mathcal{X}_T \subseteq \mathcal{E}$, whose removal yields a tree with precisely two components.

The following establishes the size of any tree-inducing cut in a graph.

**Lemma 2:** Let $G = (V, E, \mathcal{H})$ be a connected graph. The size, $X_T$, of any tree-inducing cut $\mathcal{X}_T$ in $G$ is precisely

$$X_T = |E| - |V| + 2. \quad (4)$$
Let $A$ set of linear local constraint codes.

Similarly, a graph composed of two cycle-free components satisfies

$$|E| = |V| - 2.$$  \hspace{1cm} (6)

The result then follows from the observation that the size of a tree-inducing cut is the number of edges which must be removed in order to satisfy (6).

There exists a correspondence between the edges that compose a tree-inducing cut and the cycles of that graph which establishes the following lemma.

**Lemma 3:** Let $G$ be a connected graph with tree-inducing cut size $X_T$. The number of cycles in $G$, $N_G$, is then lower-bounded by

$$N_G \geq \left(\frac{X_T}{2}\right).$$  \hspace{1cm} (7)

**Proof:** Let the removal of a tree-inducing cut $X_T$ in the connected graph $G$ yield the cycle-free components $G_1$ and $G_2$ and let $e_i \neq e_j \in X_T$. Since $G_1$ ($G_2$) is a tree, there is a unique path in $G_1$ ($G_2$) connecting $e_i$ and $e_j$. There is thus a unique cycle in $G$ corresponding to the edge pair $\{e_i, e_j\}$. There are $\left(\frac{X_T}{2}\right)$ such distinct edge pairs which yields the lower bound. Note that this is a lower bound because for certain graphs, there can exist cycles which contain more than two edges from a tree-inducing cut.

III. **Graphical Models of Codes**

Graphical models for codes have been described by a number of different authors using a wide variety of notation (e.g. [6], [7], [8], [9], [10], [11], [33]). The present work primarily uses the notation established by Forney in his *Codes on Graphs* papers [10], [33]. This section first gives a brief introduction to this notation in order to introduce $q^n$-ary graphical models in a self-contained manner.

A. **Normal Realizations of Codes**

A linear behavioral realization of a linear code $C \subseteq \mathbb{F}_q^I$ comprises three sets:

- A set of visible (or symbol) variables $\{V_i, i \in I\}$ corresponding to the codeword coordinates with alphabets $\{\mathbb{F}_q, i \in I\}$.
- A set of hidden (or state) variables $\{S_i, i \in I_S\}$ with alphabets $\{\mathbb{F}_q, i \in I_S\}$.
- A set of linear local constraint codes $\{C_j, j \in I_C\}$.

Each visible variable is $q$-ary while the hidden variable $S_i$ with alphabet $\mathbb{F}_q$ is $q^{I_S}$-ary. The hidden variable alphabet index sets $\{T_i, i \in I_T\}$ are disjoint and unrelated to $I$. Each local constraint code $C_j$ involves a certain subset of the visible, $I_V(i) \subseteq I$, and hidden, $I_S(i) \subseteq I_S$, variables and defines a subspace of the local configuration space:

$$C_j \subseteq \prod_{j \in I_V(i)} \mathbb{F}^j \times \prod_{j \in I_S(i)} \mathbb{F}^T_j.$$  \hspace{1cm} (8)

Each local constraint code $C_j$ thus has a well-defined block length $n(C_j) = |I_V(i)| + \sum_{j \in I_S(i)} |T_j|$ and dimension $k(C_j) = \dim C_j$ over $\mathbb{F}$. Local constraints that involve only hidden variables are *internal* constraints while those involving visible and hidden variables are *interface* constraints. The full behavior of the realization is the set $\mathfrak{B}$ of all visible and hidden variable configurations which simultaneously satisfy all local constraint codes:

$$\mathfrak{B} \subseteq \prod_{i \in I} \mathbb{F}^i_q \times \prod_{j \in I_S} \mathbb{F}^T_q = \mathbb{F}^I_q \times \prod_{j \in I_S} \mathbb{F}^T_q.$$  \hspace{1cm} (9)

The projection of the linear code $\mathfrak{B}$ onto $I$ is precisely $C$.

Forney demonstrated in [10] that it is sufficient to consider only those realizations in which all visible variables are involved in a single local constraint and all hidden variables are involved in two local constraints. Such *normal* realizations have a natural graphical representation in which local constraints are represented by vertices, visible variables by half-edges and hidden variables by edges. The half-edge corresponding to the visible variable $V_i$ is incident on the vertex corresponding to the single local constraint which involves $V_i$. The edge corresponding to the hidden variable $S_j$ is incident on the vertices corresponding to the two local constraints which involve $S_j$. The notation $\mathcal{G}_C$ and term *graphical model* is used throughout this work to denote both a normal realization of a code $C$ and its associated graphical representation.

In order to elucidate the graphical model properties which will be discussed in Sections III-B and III-C, the single-cycle graphical model for the length 8 extended Hamming code, $\mathcal{G}_H$, illustrated in Figure 1 will be studied throughout the remainder of this section.

![Fig. 1. Tail-biting trellis graphical model for the length 8 extended Hamming code.](image-url)

---

1Observe that this definition is slightly different than that proposed in [33] which permitted the use of $q'$-ary visible variables corresponding to $r$ codeword coordinates. By appropriately introducing equality constraints and $q$-ary hidden variables, it can be seen that these two definitions are essentially equivalent.
The tail-biting trellis model shown in Figure 1 was described in [28]. The hidden variables $S_1$ and $S_5$ are binary while $S_2$, $S_3$, $S_4$, $S_6$, $S_7$, and $S_8$ are 4-ary. All of the local constraint codes in this model are interface constraints. Equations (11)-(14) define the local constraint codes via generator matrices (where $G_i$ generates $C_i$):

\[
G_1 = \begin{bmatrix}
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad G_2 = \begin{bmatrix}
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]

\[
G_3 = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad G_4 = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

\[
G_5 = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad G_6 = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

\[
G_7 = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}, \quad G_8 = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

The complexity measure $m$ thus simultaneously captures a cyclic graphical model analog of the familiar notions of state and branch complexity for trellises [27]. The efficacy of this complexity measure is discussed further in Section V-D.

Observe that the graphical model for $C_H$ illustrated in Figure 1 is 4-ary ($q = 2$, $m = 2$). Specifically, the maximum hidden variable alphabet index set size in this model is 2 and the local constraints satisfy $\min (k(C_i), n(C_i) - k(C_i)) \leq 2$ for all $i \in I_C$. Tanner graphs are examples of $q$-ary ($m = 1$) graphical models.

The following three properties of $q^m$-ary graphical models will be used to prove the main result in Section IV:

1) Any hidden variable in a $q^m$-ary graphical model can be made to be incident on an internal local constraint, $C_i$, which satisfies $n(C_i) - k(C_i) \leq m$, without fundamentally altering the complexity or cyclic topology of that graphical model.

2) The removal of an internal local constraint from a $q^m$-ary graphical model results in a $q^m$-ary graphical model for a new code defined on same index set.

3) Any internal local constraint, $C_i$, in a $q^m$-ary graphical model satisfying $n(C_i) - k(C_i) = m'$ can be equivalently represented by $m'$ $q$-ary single parity-check equations over the visible variable index set.

These properties are discussed in detail in the following where it is assumed that a $q^m$-ary graphical model $G_C$ with behavior $B$ for a linear code $C$ over $\mathbb{F}_q$ defined on an index set $I$ is given.

1) Internal Local Constraint Involvement: Any hidden variable in a $q^m$-ary graphical model can be made to be incident on an internal local constraint, $C_i$, which satisfies $n(C_i) - k(C_i) \leq m$, without fundamentally altering the complexity or cyclic topology of that graphical model. Specifically, suppose there exists some hidden variable $S_j$, involved in the local constraints $C_{j1}$ and $C_{j2}$, that does not satisfy this property. Define a new hidden variable $S_j$ that is a copy of $S_j$ by redefining $C_{j2}$ over $S_j$ and inserting a local equality constraint $C_i$ that enforces $S_j = S_i$. The insertion of $S_j$ and $C_i$ does not fundamentally alter the complexity of the graphical model since $n(C_i) - k(C_i) = |T_j| \leq m$ and since degree-2 equality constraints are trivial from a decoding complexity viewpoint. Furthermore, the insertion of $S_j$ and $C_i$ does not fundamentally alter the cyclic topology of the graphical model since no new cycles can be introduced by this procedure.

As an example, consider the binary hidden variable $S_1$ in Figure 1 which is incident on the interface constraints $C_1$ and $C_9$. By introducing the new binary hidden variable $S_9$ and binary equality (repetition) constraint $C_9$, as illustrated in Figure 2, $S_1$ can be made to be incident on the internal constraint $C_9$.

Observe that $C_9$ satisfies the condition for inclusion in a 4-ary graphical model:

\[
n(C_9) - k(C_9) = 2 - 1 = 1 \leq 2.
\]
the generator matrices
\[
G_8 = \begin{bmatrix} 10 & 1 & 0 \\ 01 & 1 & 1 \end{bmatrix}, \quad G_9 = \begin{bmatrix} 1 & 1 \end{bmatrix}
\]  \hspace{1cm} (19)

2) Internal Local Constraint Removal: A new code on \( I \) can be defined by removing an internal local constraint from \( G_C \). Specifically, consider the removal of the internal constraint \( C_r \) in order to define \( C' \). Each hidden variable \( S_i, i \in I_S(r) \), in involved in \( C_r \) and connected to a new degree-1 internal constraint \( C_i \), which does not impose any constraint on the value of \( S_i \) (since it is degree-1). The behavior of the graphical model, \( G_{C',r} \), which results from this removal is denoted \( B' \) and the projection of \( B' \) onto \( I \) is the new code \( C' \).

As an example, consider the removal of the internal local constraint \( C_9 \) from the graphical model for \( C_H \) described in the previous section; the resulting graphical model update is illustrated in Figure 3. The new codes \( C_{10} \) and \( C_{11} \) are length 1, dimension 1 codes which thus impose no constraints on \( S_1 \) and \( S_0 \), respectively. It is readily verified that the code

\[
G_{H',r} = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.
\]  \hspace{1cm} (20)

Note that \( C'_{H,r} \) corresponds to all paths in the tail-biting trellis representation of \( C_H \) (not just those paths which begin and end in the same state).

The removal of an internal local constraint \( C_r \) results in the introduction of \( |I_S(r)| \) new degree-1 local constraints. Forney described such constraints as “useless” in [33] and they can indeed be removed from \( G_{C',r} \), since no constraints on the variables they involve. Specifically, for each hidden variable \( S_i, i \in I_S(r) \), in involved in the (removed) local constraint \( C_r \), denote by \( C_j \) the other constraint involving \( S_i \), in \( G_C \). The constraint \( C_j \) can be redefined as its projection onto \( I_V(j) \cup \{I_S(j) \setminus \} \). It can be readily verified that the resulting constraint, \( C'_{j,r} \), satisfies the condition for inclusion in a \( q^m \)-ary graphical model.

Continuing with the above example, local constraints \( C_{10} \) and \( C_{11} \), and hidden variables \( S_1 \) and \( S_0 \) can be removed from the graphical model illustrated in Figure 3 by redefining \( C_1 \) and \( C_8 \) with generator matrices

\[
G_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad G_8 = \begin{bmatrix} 10 & 1 \\ 0 & 1 \end{bmatrix}
\]  \hspace{1cm} (21)

3) Internal Local Constraint Redefinition: Any internal local constraint \( C_i \) in a \( q^m \)-ary graphical model satisfying \( n(C_i) - k(C_i) = m' \leq m \) can be equivalently represented by \( m' q \)-ary single parity-check equations over the visible variable index set. Specifically, consider a hidden variable \( S_j \) involved in \( C_i \) (i.e., \( j \in I_S(i) \)), with alphabet index set \( T_j \). Each of the \( |T_j| \) coordinates of \( S_j \) can be redefined as a q-ary sum of some subset of the visible variable set in the following manner. Consider the behavior \( B' \), and corresponding code \( C' \), which results when \( C_i \) is removed from \( G_C \) (before \( S_j \) is discarded). The projection of \( B' \) onto \( T_j \setminus I \), \( B'_{T_j \setminus I} \), has length

\[
n(C) + |T_j| \hspace{1cm} (22)
\]

and dimension

\[
k(C') \geq k(C) \hspace{1cm} (23)
\]

over \( \mathbb{F}_q \). There exists a generator matrix for \( B'_{T_j \setminus I} \) that is systematic in some size \( k(C') \) subset of the index set \( I \) [34]. Corresponding to this generator matrix is a parity-check matrix, \( H_j \), that is systematic in the \( |T_j| \) positions corresponding to the coordinates of \( S_j \). The parity-check matrix \( H_j \) thus defines each coordinate of \( S_j \) as a q-ary sum of some subset of the visible variables. In light of this observation, the \( m' q \)-ary single parity-check equations over \( I_S(i) \) which define the internal constraint \( C_i \) can be redefined.
over \( I \) by substituting into each equation the definitions of \( S_j \) implied by \( H_j \) for each \( j \in I_S(i) \).

Returning to the example of the tail-biting trellis for \( C_H \), the internal local constraint \( C_9 \), which imposes the equality constraint \( S_1 = S_9 \), can be redefined over the visible variable set as follows. The projection of \( B^9 \) (as previously defined) onto \( T_1 \cup I \) is generated by

\[
G_{B^9 | T_1, I} = \begin{bmatrix}
S_1 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
\end{bmatrix}.
\]

(24)

A valid parity check matrix for this code which is systematic over \( F \) is

\[
H_1 = \begin{bmatrix}
S_1 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 & V_8 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{bmatrix},
\]

(25)

which defines the binary hidden variable \( S_1 \) as

\[
S_1 = V_1 +_2 V_2.
\]

A similar development defines the binary hidden variable \( S_9 \) as

\[
S_9 = V_5 +_2 V_8.
\]

(27)

The local constraint \( C_9 \) thus can be redefined to enforce the single parity-check equation

\[
V_1 +_2 V_2 +_2 V_3 +_2 V_4 = 0.
\]

(28)

The three graphical model properties described above are particularly useful in concert. Specifically, suppose that the internal constraint \( C_r \) satisfying \( n(C_r) - k(C_r) = m' \leq m \) is removed from \( G_C \) resulting in the new code \( C^r \). Denote by \( C_r^{(1)}, \ldots, C_r^{(m')} \) the set of \( m' \)-ary single parity-check equations that result when \( C_r \) is redefined over \( I \). A vector in \( F_q^I \) is a codeword in \( C \) if and only if it is contained in \( C^r \) and satisfies each of these \( m' \) single parity-check equations and thus

\[
C = C^r \cap C_r^{(1)} \cap \cdots \cap C_r^{(m')}.
\]

(29)

Returning to the example of the tail-biting trellis for \( C_H \), it can readily be seen that only the first four rows of \( G_H^{10} \) (as defined in (20)) satisfy the single parity-check equation that results when \( C_9 \) is redefined over the visible variable set (as defined in (28)); it is precisely these four rows which generate \( C_H \) proving that

\[
C_H = C_H^{10} \cap C_9^{(1)}.
\]

(30)

C. The Minimal Tree Complexity of a Code

The minimal trellis complexity of a linear code \( C \) over \( \mathbb{F}_q \), \( s(C) \), is defined as the base-\( q \)-logarithm of the maximum hidden variable alphabet size in its minimal trellis [35]. Considerable attention has been paid to this quantity (cf. [31], [35], [36], [37], [38], [39]) as it is closely related to the important, and difficult, study of determining the minimum possible complexity of optimal SISO decoding of a given code. This work introduces the minimal tree complexity of a linear code as a generalization of minimal trellis complexity to arbitrary cycle-free graphical model topologies:

**Definition 4:** The minimal tree complexity of a linear code \( C \) over \( \mathbb{F}_q \) is the smallest integer \( t(C) \) such that there exists a cycle-free \( q^{t(C)} \)-ary graphical model for \( C \).

Since a trellis is a cycle-free graphical model, \( t(C) \leq s(C) \), and all known upper bounds on \( s(C) \) extend to \( t(C) \). Lower bounds on \( s(C) \) do not, however, necessarily extend to \( t(C) \). Specifically, the minimal trellis complexity of a length \( 2^m \), dimension \( m + 1 \), first-order Reed-Muller code is known to be \( m \) for \( m \geq 3 \) [36], whereas the conditionally cycle-free generalized Tanner graphs for these codes described in [40] have a natural interpretation as \( 2^{m-1} \)-ary cycle-free graphical models. The Cut-Set Bound, however, precludes \( t(C) \) from being significantly smaller than \( s(C) [10], [33] \).

An interesting question for future work is the characterization of those codes for which \( t(C) \) is strictly smaller than \( s(C) \).

The following lemma concerning minimal tree complexity is used to prove the main result in Section IV.

**Lemma 5:** Let \( C \) and \( C^{SPC} \) be linear codes over \( \mathbb{F}_q \) defined on the index sets \( I \) and \( J \subseteq I \), respectively, such that \( C^{SPC} \) is a \( q \)-ary single parity-check code. Define by \( \bar{C} \) the intersection of \( C \) and \( C^{SPC} \):

\[
\bar{C} = C \cap C^{SPC}.
\]

(31)

The minimal tree complexity of \( \bar{C} \) is then upper-bounded by

\[
t(\bar{C}) \leq t(C) + 1.
\]

(32)

**Proof:** By explicit construction of a \( q^{t(C)+1} \)-ary graphical model for \( \bar{C} \). Let \( G_\Lambda \) be some \( q^{t(C)} \)-ary cycle-free graphical model for \( C \) and let \( T \) be a minimal connected subtree of \( G_\Lambda \) containing the set of \( |J| \) interface constraints which involve the visible variables in \( J \). Denote by \( I_S(T) \subseteq I_S \) and \( I_C(T) \subseteq I_C \) the subset of hidden variables and local constraints, respectively, contained in \( T \). Choose some local constraint vertex \( C_\Lambda \), \( \Lambda \in I_C(T) \), as a root for \( T \). Observe that the choice of \( C_\Lambda \), while arbitrary, induces a directionality in \( T \); downstream toward the root vertex or upstream away from the root vertex. For every \( S_i, i \in I_S(T) \), denote by \( J_{i,1} \subseteq J \) the subset of visible variables in \( J \) which are upstream from that hidden variable edge.

A \( q^{t(C)+1} \)-ary graphical model for \( \bar{C} \) is then constructed from \( G_\Lambda \) by using each hidden variable \( S_i, i \in I_S(T) \), to also contain the \( q \)-ary partial parity of the upstream visible variables in \( J_{i,1} \subseteq J \). The local constraints \( C_j, j \in \Lambda \subseteq I_C(T) \), are updated accordingly. Finally, \( C_\Lambda \) is updated to enforce the \( q \)-ary single parity constraint defined by \( C^{SPC} \). This updating procedure increases the alphabet size of each hidden variable \( S_i, i \in I_S(T) \), by at most one and adds at most one single parity-check (or equality) constraint to the definition of each \( C_j, j \in I_C(T) \), and the
resulting cycle-free graphical model is thus at most \(q^{t(C)}+1\)-ary.

Returning for a final time to the example that began with a tail-biting trellis for \(C_H\), consider the binary graphical model for \(C_{H}^9\) illustrated in Figure 4\(^2\). All hidden variables in Figure 4 are binary and the local constraints labeled \(C_{14}, C_{17}, C_{20}, \text{and } C_{23}\) are binary single parity-check constraints while the remaining local constraints simply enforce equality. By construction, it has thus been shown that

\[
t(C_{H}) = 1. \tag{33}
\]

In light of (28) and (30), a 4-ary graphical model for \(C_{H}\) can be constructed by updating the graphical model illustrated in Figure 4 to enforce a single parity-check constraint on \(V_1, V_2, V_5, \text{and } V_6\). A minimal spanning tree containing the interface constraints incident on these variables has been highlighted in Figure 4. A natural choice for the root of this tree is \(C_{24}\). The updating of the local constraints and hidden variables contained in this spanning tree proceeds as follows. First note that since \(C_{12}, C_{15}, C_{18}, \text{and } C_{22}\) simply enforce equality, neither these constraints, nor the hidden variables incident on these constraints, need updating. The hidden variables \(S_{14}, S_{17}, S_{20}, \text{and } S_{23}\) are updated to be 4-ary so that they send downstream to \(C_{24}\) the values of \(V_1, V_2, V_8, \text{and } V_5\), respectively. These hidden variable updates are accomplished by redefining the local constraints \(C_{14}, C_{17}, C_{20}, \text{and } C_{23}\); the respective generator matrices for the redefined codes are

\[
G_{14} = \begin{bmatrix} 1 & 0 & 11 \\ 0 & 1 & 10 \end{bmatrix}, \quad G_{17} = \begin{bmatrix} 1 & 0 & 11 \\ 0 & 1 & 10 \end{bmatrix} \tag{34}
\]

\[
G_{20} = \begin{bmatrix} 1 & 0 & 11 \\ 0 & 1 & 10 \end{bmatrix}, \quad G_{23} = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 11 \end{bmatrix} \tag{35}
\]

Finally, \(C_{24}\) is updated to enforce both the original equality constraint (which corresponds to equality on the respective first coordinates of \(S_{14}, S_{17}, S_{20}, \text{and } S_{23}\)) and the additional single parity-check constraint on \(V_1, V_2, V_5, \text{and } V_6\) (which corresponds to a single parity-check on the respective second coordinates of \(S_{14}, S_{17}, S_{20}, \text{and } S_{23}\)). The generator matrix for the redefined \(C_{24}\) is

\[
G_{24} = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 01 & 00 & 00 & 01 \\ 00 & 01 & 00 & 01 \\ 00 & 00 & 01 & 01 \end{bmatrix}. \tag{36}
\]

Note that the updated constraints all satisfy the condition for inclusion in a 4-ary graphical model. Specifically, \(C_{24}\) can be decomposed into the product of a length 4 binary repetition code and a length 4 binary single-parity check code.

The updated graphical model is 4-ary and it has thus been shown by construction that

\[
t(C_H) \leq t(C_{H}^9) + 1 = 2. \tag{37}
\]

IV. The Main Result

With the required properties of \(q^m\)-ary graphical models described and Lemma 5 established, the main result can now be stated and proved.

**Theorem 6 (Tree-Inducing Cut-Set Bound):** Let \(C\) be a linear code over \(F_q\) defined on the index set \(I\) and suppose that \(G_C\) is a \(q^m\)-ary graphical model for \(C\) with tree-inducing cut size \(X_T\). The minimal tree complexity of \(C\) is then upper-bounded by

\[
t(C) \leq mX_T. \tag{38}
\]

**Proof:** By induction on \(X_T\). Let \(X_T = 1\) and suppose that \(e \in X_T\) is the sole edge in some tree-inducing cut \(X_T\) in \(G_C\). Since the removal of \(e\) partitions \(G_C\) into disconnected cycle-free components, \(G_C\) must be cycle-free and \(t(C) \leq m\) by construction.

Now suppose that \(X_T = x > 1\) and let \(e \in X_T\) be an edge in some tree-inducing cut \(X_T\) in \(G_C\). By the first \(q^m\)-ary graphical model property of Section III-B, \(e\) is incident on some internal local constraint \(C_i\), satisfying \(n(C_i) - k(C_i) = m' \leq m\). Denote by \(G_{C \setminus i}\) the \(q^m\)-ary graphical model that results when \(C_i\) is removed from \(G_C\), and by \(C^{(1)}\) the corresponding code over \(I\). The tree-inducing cut size of \(G_{C \setminus i}\) is at most \(x-1\) since the removal of \(C_i\) from \(G_C\) results in the removal of a single vertex and at least two edges. By the induction hypothesis, the minimal tree complexity of

\[\text{Fig. 4. Cycle-free binary graphical model for } C_{H}^9. \text{ The minimal spanning tree containing the interface constraints which involve } V_1, V_2, V_5, \text{and } V_6, \text{respectively, has been highlighted.}\]
Let complexi be upper-bounded by
\[ t(C^{(i)}) \leq m(x - 1). \]  
(39)

From the discussion of Section III-B, it is clear that \( C_i \) can be redefined as \( m' \leq m \) single parity check equations, \( C^{(j)} \) for \( j \in [1, m'] \), over \( \mathbb{F}_q \) on \( I \) such that
\[ C = C^{(i)} \cap C^{(1)} \cap \cdots \cap C^{(m')}. \]  
(40)

It follows from Lemma 5 that
\[ t(C) \leq t(C^{(i)}) + m' \leq mx \]  
(41)
completing the proof.

An immediate corollary to Theorem 6 results when Lemma 3 is applied to the main result:

**Corollary 7:** Let \( C \) be a linear code over \( \mathbb{F}_q \) with minimal tree complexity \( t(C) \). The number of cycles in any \( q^{m'} \)-ary graphical model for \( C \), \( N_m \), is then lower-bounded by
\[ N_m \geq \left( \frac{t(C)/m}{2} \right). \]  
(42)

**V. DISCUSSION OF THE MAIN RESULT**

**A. The Main Result and the Square-Root Bound**

The square-root bound [28] lower-bounds the complexity reduction that can be achieved by tail-biting trellises with respect to conventional trellises:

**Theorem 8 (Square-Root Bound):** Let \( C \) be a linear code over \( \mathbb{F}_q \) of even length and let \( s_{\text{mid}}(C) \) be the base-\( q \) logarithm of the minimum possible hidden variable alphabet size of a conventional trellis for \( C \) at its midpoint over all coordinate orderings. The base-\( q \) logarithm of the minimum possible hidden variable alphabet size of a tail-biting trellis for \( C \), \( s_{\text{TB}}(C) \), is then lower-bounded by
\[ s_{\text{TB}}(C) \geq s_{\text{mid}}(C)/2. \]  
(43)

Theorem 6 can be used to make a similar statement to the square-root bound which is valid for all single-cycle graphical models:

**Corollary 9:** Let \( C \) be a linear code over \( \mathbb{F}_q \) with minimal tree complexity \( t(C) \) and let \( G_C \) be a \( q^m \)-ary graphical model for \( C \) which contains a single cycle. Then
\[ m \geq t(C)/2. \]  
(44)

There are few known examples of classical codes which meet the square-root bound. Shany and Be’ery proved that many Reed-Muller codes cannot meet this bound under any bit ordering [41]. There does, however, exist a tail-biting trellis for the extended binary Golay code, \( C_G \), which meets the square root bound [28]. Specifically
\[ s_{\text{mid}}(C_G) = 8 \text{ and } s_{\text{TB}}(C_G) = 4. \]  
(45)

Given that this tail-biting trellis is a \( 2^4 \)-ary, single-cycle graphical model for \( C_G \), the minimal tree-complexity of the extended binary Golay code can thus be upper-bounded by
\[ t(C_G) \leq 8 \]  
(46)
using Corollary 9. Note that the minimal conventional trellis for \( C_G \) is a \( 2^9 \)-ary graphical model [31]. The proof of Lemma 5 provides a recipe for the construction of a \( 2^9 \)-ary cycle-free graphical model for \( C_G \) from its tail-biting trellis. It remains open as to whether the minimal tree complexity of \( C_G \) is precisely 8, however.

**B. The Main Result and the Cut-Set Bound**

The Cut-Set Bound [5], [6] is stated below without proof in the language of Section III for completeness. Note that the CSB applies to graphical models with arbitrary hidden variable size and local constraint complexities (that is, the CSB is a bound on general graphical models rather than \( q^{m'} \)-ary graphical models in particular).

**Theorem 10 (Cut-Set Bound):** Let \( C \) be a linear code over \( \mathbb{F}_q \) defined on the index set \( I \). Let \( G_C \) be a graphical model for \( C \) containing a cut \( X \) corresponding to the hidden variables \( S_i, i \in I_S(X) \), which partitions the index set into \( J_1 \subset I \) and \( J_2 \subset I \). Let the base-\( q \) logarithm of the midpoint hidden variable alphabet size of the minimal two-section trellis for \( C \) on the two-section time axis \( \{ J_1, J_2 \} \) be \( s_{X, \text{min}} \). The sum of the hidden variable alphabet sizes corresponding to the cut \( X \) is then lower-bounded by
\[ \sum_{i \in I_S(X)} |T_i| \geq s_{X, \text{min}}. \]  
(47)

On the surface, the Cut-Set Bound and Theorem 6 are similar in statement. However, there are three important differences between the two. First, the CSB does not explicitly address the complexity of the local constraints on either side of a given code. Forney provided a number of illustrative examples in [33] that stress the importance of characterizing graphical model complexity in terms of both hidden variable size and local constraint complexity.

Second, the CSB does not explicitly address the cyclic topology of the graphical model that results when the edges in a cut are removed. The removal of a tree-inducing cut results in two cycle-free disconnected components and the size of a tree-inducing cut can thus be used to make statements about the complexity of optimal SISO decoding using variable conditioning in a cyclic graphical model (cf. [10], [40], [42], [43], [44], [45]).

Finally, and most fundamentally, it difficult to use the CSB to lower bound the complexity of graphical models with cyclic topologies richer than a single cycle. Specifically, suppose the minimal complexity of a graphical model for a specific code \( C \) defined on \( J \) with a specific cyclic topology is to be determined. There are
\[ 2^{n(C) - 1} - 1 \]  
(48)
ways to partition the size \( n(C) \) visible variable index set into two non-empty, disjoint, subsets. Thus, in order to lower-bound the hidden variable complexity of \( G_C \), a potentially very large number of cuts must be considered. For each of these cuts, a different minimal two-stage trellis must be constructed in order to bound the size of that cut. What results is a linear programming bound on the minimal
hidden variable alphabet size for that specific cyclic topology. Wiberg et al. described and noted the difficulty of such an approach in the appendices of [5]. Contrast the difficulty of this CSB-based approach with one that instead utilizes the main result of the present work: provided the tree complexity of C is known (or can at least be lower-bounded), the minimal complexity of a graphical model for C with a specific cyclic topology can be readily lower-bounded using Theorem 6.

C. Asymptotic Implications of the Main Result

Denote by \( N_m \), the number of cycles in a \( q^m \)-ary graphical model for a linear code C over \( \mathbb{F}_q \) with minimal tree complexity \( t(C) \). For large values of \( t(C)/m \), the lower bound on \( N_m \) established by Corollary 7 becomes

\[
N_m \geq \left( \frac{t(C)/m}{2} \right)^2 \frac{t(C)^2}{m^2}. \tag{49}
\]

The ratio of the minimal complexity of a cycle-free model for C to that of an \( q^m \)-ary graphical model is thus upper-bounded by

\[
\frac{q^{t(C)}}{q^m} \leq 2^{2m\sqrt{N_m}}. \tag{50}
\]

Returning to the question of the limits of model extraction discussed in Section I, suppose that a simple iterative decoding algorithm for the length 255, dimension 223 Reed-Solomon code over \( \mathbb{F}_{256} \) is desired. Specifically, suppose a 16-ary graphical model for the binary image of this code, \( C_{RS} \), is desired. A reasonable estimate for the tree complexity of \( C_{RS} \) is the upper bound implied by Wolf’s bound [46]:

\[
t(C_{RS}) \approx 8(n - k) = 256. \tag{51}
\]

Assuming that (51) is accurate, the main result implies that any 16-ary graphical model for \( C_{RS} \) has a tree-inducing cut size at least 64 and thus contains at least 2016 cycles.

D. On Complexity Measures

Much as there are many valid complexity measures for conventional trellises, there are many reasonable metrics for the measurement of cyclic graphical model complexity. While there exists a unique minimal trellis for any linear block code which simultaneously minimizes all reasonable measures of complexity [47], even for the class cyclic graphical models with the most basic cyclic topology - tail-biting trellises - minimal models are not unique [29]. The complexity measure introduced by this work was motivated by the desire to have a metric which simultaneously captures hidden variable complexity and local constraint complexity thus disallowing local constraints from “hiding” complexity. There are many conceivable measures of local constraint complexity: one could upper-bound the state complexities of the local constraints or even the minimal tree complexities of the local constraints (thus defining minimal tree complexity recursively). The local constraint complexity measure used in this work is essentially Wolf’s bound [46] and is thus a potentially conservative upper bound on any reasonable measure of local constraint decoding complexity. An interesting direction for future work is the development of statements similar to the main result for alternative cyclic graphical model complexity measures.

VI. Conclusion

This work provides a novel bound - the tree-inducing cut-set bound (TI-CSB) - which characterizes of the tradeoff between cyclic topology and complexity in graphical models of linear codes. The TI-CSB provides a more precise characterization of this fundamental tradeoff than that provided by the Cut-Set Bound (CSB) and can be viewed as a generalization of the square-root bound to graphical models with cyclic topologies richer than a single cycle.

There are a number of interesting directions for future work motivated by the statement of the TI-CSB. First, while the minimal trellis complexity of linear codes is a well-understood and widely-studied quantity, less is known about the minimal tree complexity of linear codes. One direction for future work is the characterization of codes with minimal tree complexities strictly smaller than their respective minimal trellis complexities. Using the CSB to establish an upper bound on the difference between the minimal tree and trellis complexities of codes would allow for a re-expression of the TI-CSB in terms of the more familiar minimal trellis complexity. A second direction for future work is the study of codes which meet the TI-CSB with equality. Shany and Be’ery developed a necessary condition for a linear code to meet the square-root bound in [41]; developing such necessary conditions for a code to meet to the TI-CSB is an interesting open problem. A third direction for future work is the extraction of graphical models of codes for which no good graphical models are yet known. For such codes, the TI-CSB provides a metric for evaluating the quality of extracted models.

VII. Acknowledgments

This work was supported in part by the US Army Research Office under MURI Contract DAAD19-01-1-0477.

REFERENCES


